UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/829,488	04/22/2004	Lanzhong Wang	100203738-1	6921

22879 7590 03/20/2008
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LOVEL, KIMBERLY M	
ART UNIT	**PAPER NUMBER**
2167	

NOTIFICATION DATE	DELIVERY MODE
03/20/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/829,488
Filing Date: April 22, 2004
Appellant(s): WANG ET AL.

Robert Nelson, Reg. # 37,898
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 21 December 2007 appealing from the Office action mailed 16 October 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,850,538	STEINMAN	12-1998
2005/0256890	RAJASEKARAN ET AL	11-2005
6,901,207	WATKINS	5-2005

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-5 and 17-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1 and 17 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are directed toward "a computer-readable medium" and are non-statutory because they encompass subject matter and/or embodiments, which do not fall within a statutory category.

The meaning of "computer-readable medium" as disclose in the Specification, paragraph [0030], covers non-statutory embodiments which improperly include network transmission lines (interpreted as wired and wireless transmission), wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Paragraph [0030] of the specification provides evidence that applicant intends the medium to include signals as such the claim is drawn to a form of energy. Energy is not one of the four categories of invention and therefore the claims are not statutory. Energy is not a series of steps or acts and thus is not a process. Energy is not a physical article and as such is not a machine or manufacture. Energy is not combination of substances and therefore not a composition of matter.

Claims 2-5 and 18-20, which are dependent respectively on claims 4 and 18 fail to overcome the rejection and therefore are rejected on the same grounds as claims 4 and 18.

To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. 101 (nonstatutory) above are further rejected as set forth below in anticipation of applicant amending these claims to place them within the four statutory categories of invention.

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-4, 6-15, 17-19 and 21-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No 5,850,538 to Steinman (hereafter Steinman) in view of US PGPub 2005/0256890 to Rajasekaran et al (hereafter Rajasekaran).

Referring to claim 1, Steinman discloses a data structure [queue data structure] comprising:

a sorted portion that contains a plurality of entries that are sorted into an order [main priority queue] (see column 6, line 65 – column 7, line 4);

an unsorted portion that contains a plurality of entries that have not been sorted [temporary holding queue] (see column 6, line 65 – column 7, line 4); and

a boundary that separates the sorted portion and the unsorted portion [two separate lists] (see column 6, line 65 – column 7, line 9).

an entry is added to the unsorted portion [temporary queue] (see column 6, line 65 – column 7, line 9).

However, Steinman fails to explicitly disclose the further limitation wherein the sorted portion of the data structure is searchable with O(logN) performance. Rajasekaran discloses sorted lists, including the further limitation wherein the sorted portion [sorted list] of the data structure is searchable with O(logN) performance (see [0123]) in order to provide fast and efficient search techniques.

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the concept of searching the sorted portion of the list using a binary search as disclosed by Rajasekaran while adding data entries to the unsorted portion as disclosed by Steinman. One would have been motivated to do so in order to provide fast and efficient search techniques (Rajasekaran: see [0004]).

Referring to claim 2, the combination of Steinman and Rajasekaran (hereafter Steinman/Rajasekaran) discloses the data structure of claim 1, wherein the sorted portion is searchable with a binary search [O(log N)] (Rajasekaran: see [0123]).

Referring to claim 3, Steinman/Rajasekaran discloses the data structure of claim 1, wherein the unsorted portion is searchable with an incremental search (Steinman: see column 9, lines 31-36).

Referring to claim 4, Steinman/Rajasekaran discloses the data structure of claim 1, wherein the data structure may be sorted to form a new sorted portion that comprises the plurality of entries of the sorted portion [main priority queue] and the plurality of entries of the unsorted portion [temporary holding queue], and the plurality of entries of the new sorted portion are sorted into an order [temporary queue is sorted and merged the main priority queue] (Steinman: see column 7, lines 1-18).

Referring to claim 6, Steinman discloses a method of using a container that comprises a sorted portion [main priority queue] that contains a plurality of entries that are sorted into an order, an unsorted portion that contains a plurality of entries that have not been sorted [temporary holding queue], and a boundary that separates the sorted portion and the unsorted portion [two separate lists] (see column 6, line 65 – column 7, line 9), the method comprising:

receiving a search request that comprises a requested value (see column 6, line 65 – column 7, line 9);

searching the sorted portion of the container for the requested value with O(logN) performance (see column 6, line 65 – column 7, line 9);

adding an entry to the unsorted portion [temporary queue] during the searching (see column 7, lines 3-4);

and returning a stored value of the container if there is a match of the stored value and the requested value (see column 6, line 65 – column 7, line 9).

However, Steinman fails to explicitly disclose the further limitation wherein the sorted portion of the data structure may be searched with O(logN) performance. Rajasekaran discloses sorted lists, including the further limitation wherein the sorted portion [sorted list] of the data structure may be searched with O(logN) performance (see [0123]) in order to provide fast and efficient search techniques.

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the concept of searching the sorted portion of the list using a binary search as disclosed by Rajasekaran while adding data entries to the unsorted portion as disclosed by Steinman. One would have been motivated to do so in order to provide fast and efficient search techniques (Rajasekaran: see [0004]).

Referring to claim 7, Steinman/Rajasekaran discloses the method of claim 6, wherein when there is not a match, the method further comprises: returning a null value that indicates that there is no match with the requested value (Steinman: see column 7, lines 1-18).

Referring to claim 8, Steinman/Rajasekaran discloses the method of claim 6, wherein when there is not a match, the method further comprises: adding an entry to the unsorted portion corresponding to the search request (Steinman: see column 7, lines 3-4).

Referring to claim 9, Steinman/Rajasekaran discloses the method of claim 6, wherein when there is not a match, the method further comprises:

determining whether unsorted items in the container exceed a predetermined threshold [the event horizon has been crossed] (Steinman: see column 7, lines 1-18);

performing a sort operation on the container, if the predetermined threshold is exceeded, thereby forming a new sorted portion that comprises the plurality of entries of the sorted portion [main priority queue] and the plurality of entries of the unsorted portion [temporary holding queue], and the plurality of entries of the new sorted portion are sorted into an order [temporary queue is sorted and merged with the main priority queue] (Steinman: see column 7, lines 1-18).

Referring to claim 10, Steinman/Rajasekaran discloses the method of claim 9, further comprises: searching the new sorted portion of the container for the requested value; and returning a stored value of the container if there is a match of the stored value and the requested value (Rajasekaran: see [0123]).

Referring to claim 11, Steinman/Rajasekaran discloses the method of claim 10, wherein searching the new sorted portion comprises: searching with O(logN) performance (Rajasekaran: see [0123]).

Referring to claim 12, Steinman/Rajasekaran discloses the method of claim 6, wherein when there is not a match, the method further comprises: searching the unsorted portion of the container for the requested value; and returning a stored value of the container if there is a match of the stored value and the requested value (Steinman: see column 9, lines 31-36).

Referring to claim 13, Steinman/Rajasekaran discloses the method of claim 12, wherein the unsorted portion may be searched with an incremental search [O(n)] (Steinman: see column 9, lines 31-34).

Referring to claim 14, Steinman/Rajasekaran discloses the method of claim 6, wherein when there is not a match, the method further comprises: determining whether a size of the unsorted portion is zero; adding an entry to the unsorted portion corresponding to the search request if the unsorted portion is not zero (Steinman: see column 7, lines 1-18).

Referring to claim 15, Steinman/Rajasekaran discloses the method of claim 14, wherein the size of the unsorted portion is zero, the method further comprises: determining whether the requested value of the search request is greater than the value of the last entry of the sorted portion; adding an entry to the unsorted portion corresponding to the search request if the requested value of the search request is not greater than the value of the last entry of the sorted portion; adding an entry to the sorted portion corresponding to the search request if the requested value of the search request is greater than the value of the last entry of the sorted portion (Steinman: see column 7, lines 1-18).

Referring to claim 17, Steinman discloses a computer program product having a computer-readable medium having computer program logic recorded thereon for inserting a new value into a container that comprises a sorted portion [main priority queue] that contains a plurality of entries that are sorted into an order, an unsorted portion [temporary priority queue] that contains a plurality of entries that have not been sorted, and a boundary that separates the sorted portion and the unsorted portion (see column 9, lines 31-36), the computer program product comprising:

code for searching the sorted portion of the container for the new value with O(logN) performance (see column 6, line 65 – column 7, line 9 and column 9, lines 31-36);

code for searching the unsorted portion of the container if no match is found in the search of the sorted portion with O(N) performance (see column 9, lines 31-36); and

code for inserting the new value into the container if no match is found in the search of the unsorted portion (see column 6, line 65 – column 7, line 9).

However, Steinman fails to explicitly disclose the further limitation wherein the sorted portion of the data structure may be searched with O(logN) performance. Rajasekaran discloses sorted lists, including the further limitation wherein the sorted portion [sorted list] of the data structure may be searched with O(logN) performance (see [0123]) in order to provide fast and efficient search techniques.

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the concept of searching the sorted portion of the list using a binary search as disclosed by Rajasekaran while adding data entries to the unsorted portion as disclosed by Steinman. One would have been motivated to do so in order to provide fast and efficient search techniques (Rajasekaran: see [0004]).

Referring to claim 18, Steinman/Rajasekaran discloses the computer program product of claim 17, wherein the code for inserting comprises: code for determining whether to insert the new value in the sorted portion or the unsorted portion of the container (Steinman: column 6, line 65 – column 7, line 9 and column 9, lines 31-36 and see column 9, lines 31-36).

Referring to claim 19, Steinman/Rajasekaran discloses the computer program product of claim 17, further comprises: code for sorting the unsorted portion and merging the sorted portion and the sorted unsorted portion into a new sorted portion, wherein the code for sorting is operative when the unsorted portion exceeds a predetermined criteria [the event horizon has been crossed] (Steinman: see column 7, lines 1-18); and code for searching the new sorted portion of the container for the new value with O(logN) performance (Rajasekaran: see [0123]).

Referring to claim 21, Steinman/Rajasekaran discloses a computer system for managing data objects, comprising:

memory means for storing said data objects [main priority queue] (see column 6, line 65 – column 7, line 4);

means for identifying a boundary within said memory means for storing, wherein data objects stored in a first portion [main priority queue] of said memory means defined by said boundary are stored in an ordered manner [sorted] and data objects stored in a second portion [temporary holding queue] of said memory means defined by said boundary are stored in an unordered manner [two separate queues] (see column 6, line 65 – column 7, line 4); and

means for searching said first portion according to O(logN) performance [O(log N)] to locate an identified object (Steinman: see column 9, lines 31-36).

Referring to claim 22, Steinman/Rajasekaran discloses the computer system of claim 21 further comprising: means for searching said second portion for said identified object according to O(N) performance (Steinman: see column 9, lines 31-36).

Referring to claim 23, Steinman/Rajasekaran discloses the computer system of claim 21 further comprising: means for adding said identified object to said second portion when said means for searching said first portion and said means for searching said second portion do not locate said identified object (Steinman: see column 7, lines 1-18).

Referring to claim 24, Steinman/Rajasekaran discloses the computer system of claim 21 further comprising: means for merging data objects in said second portion [temporary holding queue] into said first portion [main priority queue] in an ordered manner; and means for resetting said boundary in response to said means for merging (Steinman: see column 7, lines 1-18).

Referring to claim 25, Steinman/Rajasekaran discloses the computer system of claim 24 wherein said means for merging is operable when a number of data objects in said second portion reaches a predetermined amount [the event horizon has been crossed] (Steinman: see column 7, lines 3-4).

Claims 5, 16 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No 5,850,538 to Steinman in view of US PGPub 2005/0256890 to Rajasekaran et al as applied to respectively to claims 1, 6 and 17 above, and further in view of US Patent No 6,901,207 to Watkins (hereafter Watkins).

Referring to claim 5, Steinman/Rajasekaran discloses a data structure. However, Steinman/Rajasekaran fails to explicitly disclose the further limitation wherein the data structure is associated with an occurrence model used in designing a circuit. Watkins discloses a data structure, including the further limitation of wherein the data structure is associated with an occurrence model used in designing a circuit (Watkins: see column 19, lines 50-57) so in order to improve the efficiency of designing the circuit.

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the data structure of Steinman/Rajasekaran with the occurrence model of Watkins. One would have been motivated to do so in order to improve the efficiency of designing the circuit.

Referring to claims 16 and 20, the claims are rejected on the same grounds as claim 5.

(10) Response to Argument

This Examiner's Answer will address the Appellants' arguments in the order in which they appear in the appeal brief.

- ***Issue I:* The Rejection of Claims 1-5 and 17-20 under 35 USC § 101**

Appellants' Arguments: The office action states that the claimed invention is directed toward non-statutory subject matter. In summary, the office action further states that the claims are directed toward forms of energy, which the office action states are not patentable. Claim 1 is directed toward a data structure stored on a computer-readable medium. Thus, claim 1 is not directed toward a form of energy as set forth in the final office action. Rather, claim 1, is directed toward a computer-readable medium with a data structure stored thereon (Appeal Brief: pages 6-8).

Examiner's Response: The examiner agrees that the claim is directed towards a data structure. However, according to the MPEP, a data structure, which is construed as functional descriptive material per se, will be statutory in most cases when it is recorded on some computer-readable medium. The examiner does not contest that claim 1 is directed towards a data that is stored on a computer-readable medium.

However, in this instance, the computer-readable medium as disclosed by the Appellants' specification ([0030]) encompasses subject matter which is considered to be non-statutory subject matter.

MPEP 2106.01 [R-6]:

> Both types of “descriptive material” are nonstatutory when claimed as descriptive material per se, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare In re Lowry, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (discussing patentable weight of data structure limitations in the context of a statutory claim to a data structure stored on a computer readable medium that increases computer efficiency) and >In re< Warmerdam, 33 F.3d *>1354,< 1360-61, 31 USPQ2d *>1754,< 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with Warmerdam, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory).

Appellants' Specification – paragraph [0030]

[0030] When implemented in software, the elements of the present invention are essentially the code segments to perform the necessary tasks. The program or code segments can be stored in a processor-readable medium or transmitted by a computer data signal embodied in a carrier wave, or a signal modulated by a carrier, over a transmission medium. **The "processor-readable medium" may include any medium that can store or transfer information**. Examples of the processor-readable medium include an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable ROM (EROM), a floppy diskette, a compact disk CD-ROM, an optical disk, a hard disk, a fiber optic medium, a radio frequency (RF) link, etc. The computer data signal may include any signal that can propagate over a transmission medium such as electronic network channels, optical fibers, air, electromagnetic, RF links, etc. The code segments may be downloaded via computer networks such as the Internet, intranet, etc.

Referring to claim 16, the examiner disagrees that the computer-readable medium is directed towards statutory subject matter. The computer-readable medium as disclosed by the Appellants' specification ([0030]) encompasses subject matter which is considered to be non-statutory subject matter.

- ***Issue II:* The Rejection of Claims 1-4, 6-15, 17-19 and 21-25 under 35 USC § 103(a)**

Appellants' Arguments: Claim 1 includes a data structure comprising two portions, a sorted portion and an unsorted portion. In summary, the Appellants contend that Steinman does not disclose a single data structure comprising two portions as claimed. According to the office action, these portions are disclosed by Steinman at column 6, line 65 to column 7, line 9. Based on this section of Steinman, a "main priority queue data structure" stores sorted data and an "unsorted temporary hold queue" stores unsorted data. Thus, sorted data is stored in one data structure and unsorted data is stored in the temporary holding queue, which is different than the main priority queue. Therefore, Steinman does not disclose a data structure comprising the sorted portion and the unsorted portion as claimed in claim 1. Rather, there are two different data structures (Appeal Brief: pages 9-10).

Examiner's Response: The examiner respectfully disagrees that the term "a data structure" has to be explicitly limited to one data structure. The specification fails to explicitly define or disclose the type of data structure. Therefore, according to MPEP 2106 [R-5], "USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure." Also, a data structure can be recursive meaning that the data structure is partially composed of smaller instances of the same data structure. Steinman discusses a data structured referred to as a Qheap, which is a heap using only linked lists ("Instead of describing heaps using binary trees and exploiting modular arithmetic schemes with fixed arrays, the present invention is embodied in a novel implementation of a heap using only linked lists. The new data structure of the present invention is the preferred embodiment and is referred to as a Qheap." – column 16, lines 1-6). Steinman further discloses a specialized Qheap referred to as a SPEEDES Qheap, which is made up of Q [the sorted portion] and Q_{temp} [the unsorted portion] ("Further, the event horizon can be applied to the Qheap to form a SPEEDES Qheap in accordance with the present invention. FIG. 12 is a flow chart illustrating an overview of the Qheap operation with the event horizon. Namely, instead of directly inserting events into Q, they are added to Q.sub.temp. When the event horizon is crossed, Q.sub.temp is sorted, the top item is removed as the next event, the rest of the list is metasized, and then inserted into Q. The obvious advantage of using Q.sub.temp is to provide for larger numbers of events to be in a single metaitem, thereby reducing the average number of untangling steps. This new data structure is an alternative embodiment of the Qheap (the preferred embodiment) of the present invention and will be referred to as the SPEEDES Qheap, and can be used in the SPEEDES operating system.") (column 17, lines 42-56). Also, it is well known to one of ordinary skill in the art that linked lists are considered to represent a single data structure. Interpreting the data structure as being limited to a single data structure as opposed to being a data structure that is comprises multiple smaller data structures is merely a matter of semantics. Therefore, it is the examiner's position that the main priority queue [Q] and the temporary holding queue [Qtemp] meet the claim's requirement of a data structure comprising a sorted portion and an unsorted portion.

Appellants' Arguments: The office action refers to the event horizon of Steinman as being the boundary of claim 1. Again, the applicants disagree with this holding. The event horizon taught by Steinman is a time stamp and is not a boundary as claimed in claim 1. (Appeal Brief: page 10).

Examiner's Response: It is noted that the Office Action does not refer to the event horizon as being equivalent to the boundary. As stated above in the rejection of claim 1 and as stated previously on pages 4 and 13-16 of the Office Action dated 16 October 2007, the boundary is considered to be represented by the concept of the existence of two lists. The claim limitation states "a boundary that separates the sorted portion from the unsorted portion." Paragraph [0016] of Appellants' specification states "The boundary 130 marks the division of the container 100 between the sorted and unsorted portions." The cited portion of Steinman states "In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, they are not immediately sent back into the main priority queue data structure, but instead are collected in an unsorted temporary holding queue. The event with the earliest time in tag in this queue is tracked. When the next event to be processed is in the temporary queue (i.e., the event horizon has been crossed), the queue is sorted (a binary merge sort algorithm is easily performed on linked lists) and then merged back into the main priority queue data structure" (Steinman: see column 6, line 65 – column 7, line 8). According to MPEP 2106 [R-5], "USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure." The examiner is interpreting the term "boundary" to mean a line, space or division forming a separation between the sorted portion and the unsorted portion. This interpretation coincides with the definition in the Appellants' specification and with how one of ordinary skill in the art would define the term. Since Steinman discloses a separate main priority queue [sorted portion] and a temporary queue [unsorted portion], there is inherently a division that exists between the two queues. Therefore, the examiner respectfully disagrees that Steinman fails to disclose a boundary.

Appellants' Arguments: Claim 1 also recites "wherein the sorted portion of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion." According to the office action, Steinman discloses adding an entry to the unsorted portion, but there is no mention of adding the entry while searching the sorted portion as recited in claim 1. The cited portions of Steinman disclose searching a list and adding an item as separate functions, not together (Appeal Brief: page 9-11).

Examiner's Response: The Appellants' argument is centered around claim interpretation. It appears that the Appellants are interpreting the term "while" to mean simultaneously where as the examiner is interpreting the term as being used in the contrastive sense of although or where as. Paragraph [0015] of Appellants' specification states "During operations using the data structure, the sorted portion of the data structure may be searched in accordance with the boundary for O(logN) performance, while new items can be added into the unsorted portion of the container to be sorted and merged into the sorted portion later" and paragraph [0017] states "A binary search with O(logN) performance, may be performed on the sorted portion 110, even though unsorted items have been inserted or are present in the container." Therefore, the Appellants' specification fails to explicitly limit the meaning of the term "while" to mean simultaneously. According to MPEP 2106 [R-5], "USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure." Steinman is utilized to teach the portion of the limitation that states "an entry is added to the unsorted portion ("In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, they are not immediately sent back into the main priority queue data structure, but instead are collected in an unsorted temporary holding queue …" – column 6, line 65 – column 7, line 9). Rajasekaran is utilized to teach the portion of the claim that states "wherein the sorted portion of the data structure is searchable with O(logN) performance." Therefore, the combination of

Steinman and Rajasekaran is considered to meet the requirements of the limitation, when the limitation is given its broadest reasonable interpretation.

Appellants' Arguments: As stated above, neither Steinman nor Rajasekaran disclose the boundary as claimed. Rather, Steinman discloses an event horizon that functions as a time stamp and does not boundaries as claimed. Furthermore and as stated above, neither reference discloses adding an entry to the unsorted of a container during a search of the sorted portion. (Appeal Brief in reference to claim 6: pages 11-12)

Examiner's Response: Referring to the first argument, it is noted that the Office Action does not refer to the event horizon as being equivalent to the boundary. As stated above, the boundary is considered to be represented by the concept of the existence of two lists. The claim limitation states "a boundary that separates the sorted portion from the unsorted portion." Paragraph [0016] of Appellants' specification states "The boundary 130 marks the division of the container 100 between the sorted and unsorted portions." The cited portion of Steinman states "In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, they are not immediately sent back into the main priority queue data structure, but instead are collected in an unsorted temporary holding queue. The event with the earliest time in tag in this queue is tracked. When the next event to be processed is in the temporary queue (i.e., the event horizon has been crossed), the queue is sorted (a binary merge sort algorithm is easily performed on linked lists) and then merged back into the main priority queue data structure" (Steinman: see column 6, line 65 – column 7, line 8). According to MPEP 2106 [R-5], "USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure." The examiner is interpreting the term "boundary" to mean a line, space or division forming a separation between the sorted portion and the unsorted portion. This interpretation coincides with the definition in the Appellants' specification and with how one of ordinary skill in the art would define the term. Since Steinman discloses a separate main priority queue [sorted portion] and a temporary queue [unsorted portion], there is inherently a division that exists between the two queues. Therefore, the examiner respectfully disagrees that Steinman fails to disclose a boundary.

Referring to the second argument, the Appellants' argument is centered around claim interpretation. It appears that the Appellants are interpreting the term "while" to mean simultaneously where as the examiner is interpreting the term as being used in the contrastive sense of although or where as. Paragraph [0015] of Appellants' specification states "During operations using the data structure, the sorted portion of the data structure may be searched in accordance with the boundary for O(logN) performance, while new items can be added into the unsorted portion of the container to be sorted and merged into the sorted portion later" and paragraph [0017] states "A binary search with O(logN) performance, may be performed on the sorted portion 110, even though unsorted items have been inserted or are present in the container."

Therefore, the Appellants' specification fails to explicitly limit the meaning of the term "while" to mean simultaneously. According to MPEP 2106 [R-5], “USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure.” Steinman is utilized to teach the portion of the limitation that states “an entry is added to the unsorted portion ("In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, they are not immediately sent back into the main priority queue data structure, but instead are collected in an unsorted temporary holding queue …” – column 6, line 65 – column 7, line 9). Rajasekaran is utilized to teach the portion of the claim that states “wherein the sorted portion of the data structure is searchable with O(logN) performance.” Therefore, the combination of Steinman and Rajasekaran is considered to meet the requirements of the limitation, when the limitation is given its broadest reasonable interpretation.

Appellants’ Arguments: As stated above, neither Steinman nor Rajasekaran disclose the boundary as claimed. Rather, Steinman discloses an event horizon that functions as a time stamp and does not boundaries as claimed. (Appeal Brief in reference to claim 17: pages 12-13)

Examiner’s Response: It is noted that the Office Action does not refer to the event horizon as being equivalent to the boundary. As stated above, the boundary is considered to be represented by the concept of the existence of two lists. The claim limitation states "a boundary that separates the sorted portion from the unsorted portion." Paragraph [0016] of Appellants' specification states "The boundary 130 marks the division of the container 100 between the sorted and unsorted portions." The cited portion of Steinman states "In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, they are not immediately sent back into the main priority queue data structure, but instead are collected in an unsorted temporary holding queue. The event with the earliest time in tag in this queue is tracked. When the next event to be processed is in the temporary queue (i.e., the event horizon has been crossed), the queue is sorted (a binary merge sort algorithm is easily performed on linked lists) and then merged back into the main priority queue data structure" (Steinman: see column 6, line 65 – column 7, line 8). According to MPEP 2106 [R-5], "USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure." The examiner is interpreting the term "boundary" to mean a line, space or division forming a separation between the sorted portion and the unsorted portion. This interpretation coincides with the definition in the Appellants' specification and with how one of ordinary skill in the art would define the term. Since Steinman discloses a separate main priority queue [sorted portion] and a temporary queue [unsorted portion], there is inherently a division that exists between the two queues. Therefore, the examiner respectfully disagrees that Steinman fails to disclose a boundary.

Appellants' Arguments: As stated above, neither Steinman nor Rajasekaran disclose the boundary as claimed. (Appeal Brief in reference to claim 21: pages 12-13)

Examiner's Response: It is noted that the Office Action does not refer to the event horizon as being equivalent to the boundary. As stated above, the boundary is considered to be represented by the concept of the existence of two lists. The claim limitation states "a boundary that separates the sorted portion from the unsorted portion." Paragraph [0016] of Appellants' specification states "The boundary 130 marks the division of the container 100 between the sorted and unsorted portions." The cited portion of Steinman states "In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, they are not immediately sent back into the main priority queue data structure, but instead are collected in an unsorted temporary holding queue. The event with the earliest time in tag in this queue is tracked. When the next event to be processed is in the temporary queue (i.e., the event horizon has been crossed), the queue is sorted (a binary merge sort algorithm is easily performed on linked lists) and then merged back into the main priority queue data structure" (Steinman: see column 6, line 65 – column 7, line 8). According to MPEP 2106 [R-5], "USPTO personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure." The examiner is interpreting the term "boundary" to mean a line, space or division forming a separation between the sorted portion and the unsorted portion. This interpretation coincides with the definition in the Appellants' specification and with how one of ordinary skill in the art would define the term. Since Steinman discloses a separate main priority queue [sorted portion] and a temporary queue [unsorted portion], there is inherently a division that exists between the two queues. Therefore, the examiner respectfully disagrees that Steinman fails to disclose a boundary.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Kimberly Lovel/

Kimberly Lovel
Examiner
2167

Conferees:

/Eddie Lee/
Supervisory Patent Examiner, TC 2100

Eddie Lee
TQAS/Appeals Specialist
TC 2100

John Cottingham
Supervisor Patent Examiner
AU 2167
/John R. Cottingham/

Supervisory Patent Examiner, Art Unit 2167